

The advantages of parallel processing used to measure the performance of image processing on led walls on a national Albanian media

Nevila Baci ^{a1}, University of Tirana Faculty of Economy, Albania.

Denisa Millo ^b, University of Tirana Faculty of Economy, Albania.

Suggested Citation:

Baci, N. & Millo, D. (2022). The advantages of parallel processing used to measure the performance of image processing on led walls on a national Albanian media. *Global Journal of Computer Sciences: Theory and Research*. 12(1), 55-61. <https://doi.org/10.18844/gjcs.v12i1.7448>

Received from December 17, 2021; revised from February 25, 2022; accepted from April 11, 2022.

Selection and peer-review under the responsibility of Assist. Prof. Dr. Ezgi Pelin Yıldız, Kafkas University.

©2022 Birlesik Dunya Yenilik Arastirma ve Yayıncılık Merkezi. All rights reserved.

Abstract

The use of AI technology has led to efficiency and better performance, as seen from the point of view of an economist, as efficiency may bring about, and thus better revenues. But the AI technology is also supplied by some important technologies such as parallel processing. This article aims to analyze the benefits of parallel processing by illustrating it with a code in C, using the MPI library. The experiment is done for the first time for academic purposes in a High-Performance Computer used in a university space, and then re-applied in the business space of the media which uses the latter kind of computer with some more advanced features to process the images in led walls. The results show that the image is the better and faster process when the number of nodes increases, but the way it behaves depends on the relation between the number of processes and the number of processors in the nodes.

Keywords: Image processing; High-Performance Computing; Parallel Processing; media; MPI.

* ADDRESS FOR CORRESPONDENCE: Nevila BACI* University of Tirana Faculty of Economy, Albania.
E-mail address: Nevila.baci@unitir.edu.al

1. Introduction

Nowadays the role of the computer is evidenced in every aspect of human activity. Besides its known advantages, relies on the most important usage, and time shortage of processing compared to other ways of human technology. Taking into consideration the need for managing better the time it is important to develop new ways. Parallel processing is such that new way to improve in time with the help of today's computers. In a specific definition by Oxford Dictionary [1] parallel processing is "a mode of operation in which a process is split into parts, which are executed simultaneously on different processors attached to the same computer."

1.1. Literature review

1.1.1. The economic approach to parallel processing

The parallel processing reaches out to take a job in a shorter time, dividing it into pieces. This concept is rooted in the human mind since he first used reason. Being alone, he couldn't make any of the ordinary works that were the source of living. Living in groups was brought about by the necessity of surviving, which made possible the division of labor.

The father of classic liberalism [] has analyzed the parallelism concept relating to labor division. In his famous book is cited the importance of labor division. We can deduce therefore the parallelism importance as well. It is said, "The biggest improvement of production power and a good part of ability, mastery, and intelligence that it is accompanied by, seemed to have generated from labor division". The principles are listed below.

First, the improvement of workers' mastery (let us make an analogy with computer or application) in one way or another adds the amount of work they can do. According to the first principle, it is obvious that in a case an application is reduced to small pieces, any one of which can be processed in every processor or computer, then the time of processing will be highly improved.

The second principle deals with "The gained advantage of the time that is won during the transition from work to work is greater than the one that can be judged from the first sight". To bring it in informatics terms we can understand as the time saved the worker wouldn't have to travel from one computer to another. The time would be less if that computer/processor is specialized in one simple work.

The third principle relates to the improvement of processing time because of the usage of technology [3-6]. We can remind here that it was the period of the Industrial Revolution when machinery brought about visible changes in different aspects of life, especially in the economy.

In the following article, we will analyze the parallel processing used to process the images in led walls for a media, its advantages, and disadvantages.

1.1.2. Computer Approach. On the functional principle of parallel processing

In its simplest meaning, parallel processing is the simultaneous usage of computing sources to solve a computer problem. To be done such processing there have to be used multiple CPU-s for a problem that is divided into discrete parts can be solved at the same time. Every part is then processed by serial instructions.

In [3] are explained the reasons for using parallel processing more and more. There it is cited "The quest for higher-performance digital computers seems unending. In the past two decades, the performance of microprocessors has enjoyed exponential growth. The growth of microprocessor speed/performance by a factor of 2 every 18 months (or about 60% per year) is known as Moore's law. This growth is the result

of a combination of two factors:

1. An increase in complexity (related both to higher device density and to larger size) is projected to rise to around 10M transistors per chip for microprocessors, and 1B for dynamic random-access memories (DRAMs), by the years 2000[SIA94]
2. Introduction of, and improvements in, architectural features such as on-chip cache memories, large instruction buffers, multiple instruction issues per cycle, multithreading, deep pipelines, out-of-order instruction execution, and branch prediction. In our work, to explain parallel processing, we are going to use MPI (message-passing-interface) library. There exist different packages and languages that can help make a program in parallel. The one we are considering is one of the most used.

In the treatment [3] makes for the MPI it is said that the Message Passing interface is a specification for developers and users of libraries to transfer the messages. It is NOT a bookstore, but rather a specification of how a library should be. The goals of MPI are to provide a wide standard for programs to be written on how to transfer messages. MPI is a bridge of communication between processes that share different address spaces. Interface aims to be practical, efficient, flexible, and portable. It must be said that specifications are defined for programs written in languages C / C + + and Fortran. *What are the reasons for using APIs?*

It must be mentioned here is the portability. The poor availability of source code modified when transferred to another platform or application that supports MPI is the main advantage of this library. Association with opportunities for optimal performance makes MPI have widespread usage. Another reason we could enumerate is the functionality, considering that over 115 routines are defined in MPI-1. A large variety of public and private implementations of MPI give the advantage of widespread availability.

1.2. Related research

The question raised in this article is as follows: "How do the image processing indicators such as time of processing, improve by applying it in parallel, and how did they change from having one node with 8 processors to two nodes with 16 processors until reaching eighteen nodes with 144 processors?". In the works of [5-9], we can find similar usage of parallelism and its broad applications in different fields of science.

1.3. Purpose of study

The emphasis of this article is on analyzing the benefits of parallel processing by illustrating it with a code in C using the MPI library. The latter use is analyzed in the context of image processing in a led wall platform of a national media, which supports a recent technology of parallel processing.

2. Materials and Methods

2.1. Data collection

The research is experimental. The program is processed in different ways in parallel. The experiment is done for the first time for academic purposes in a High-Performance Computer used in a university space, and then re-applied in the business space of the media which uses the latter kind of computer with some more advanced features to process the images in led walls. In genesis, MPI was designed for distributed memory but today it can be used for shared memory and both. The led walls in our media use the shared-memory type of memory. Because the image comes from some different sources and is then gathered to form an integrated image that can be understood by the public.

2.2. Analysis

2.2.1. The role of MPI and the results of parallel processing tests

First let us bring the code of the program where we want to find the sine, cosine, and tangent of a certain angle which is used in image processing of the brightness angle of the image. This program is integrated within the software of image processing of the media HPC and is processed under strict supervision when taking the results and their effect on image processing. The code is taken used also in the published paper [5].

```
#include <mpi.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
typedef struct
{double thred;
double start;
double stop;
double data1;
double data2;
double data3;}
trigonom;
trigonom * array;
trigonom procs;
int main (int argc, char **argv)
{int rank, size, nrpr, i, j, k;
double sinsin, coscos, tantan, fillo, mbaro;
long in there;
nrpr = 1; here=222;
if (argc>0) nrpr = atoi (argv [1]);
// numri i proceseve
// fillo MPI
MPI_Init (&argc, &argv);
MPI_Comm_rank (MPI_COMM_WORLD,
&rank);
MPI_Comm_size (MPI_COMM_WORLD, &size);
// printon inputet dhe merr kohen ne procesin
meme
if (rank==0)
{fillo = MPI_Wtime ();
printf ("input nrpr=%d \nprint);
array = malloc (nrpr*sizeof(trigonom));
for (i=0; i<nrpr; i++)
{array[i]. thred=0;
array[i]. start=0;
array[i]. stop =0;
array[i].data1=0;
array[i].data2=0;
array[i].data3=0;}}
MPI_Scatter(&array[rank], 6, MPI_DOUBLE,
&procs,6, MPI_DOUBLE,0,
MPI_COMM_WORLD);
// ekzekutohen loopet e brendshme te
proceseve
procs.start = MPI_Wtime ();
for (j=0; j<720720/nrpr;j++)
for (k=0; k<here; k++)
{sinsin=sin(k);
coscos=cos(k);
tantan=tan(k);}
procs.thred= rank;
procs.data1=sinsin;
procs.data2=coscos;
procs.data3=tantan;
procs.stop = MPI_Wtime();
MPI_Gather(&procs,6,MPI_DOUBLE,array,6,MPI
_DOUBLE,0,MPI_COMM_WORLD);
// printohet vektori runtime i procesit ne
procesin meme
if(rank==0)
{for (i=0; i<nrpr; i++)
printf ("Iter %d process %f Runtime %f \n",
i,array[i].thred,array[i].stop -array[i].start);
// printohet koha e ekzekutimit
mbaro = MPI_Wtime();
printf("time = %f \n",mbaro-fillo);}
// mbyllet MPI
MPI_Finalize();
return 0;}
```

The key concept of the program code above stands in this diagram. The MPI_Scatter routine philosophy sends the data from a specific address to the structure defined in the code so they can be calculated. After

the calculation, the MPI_gather routine gathers the information from the procs structure after being processed and puts it in the beginner structure again.

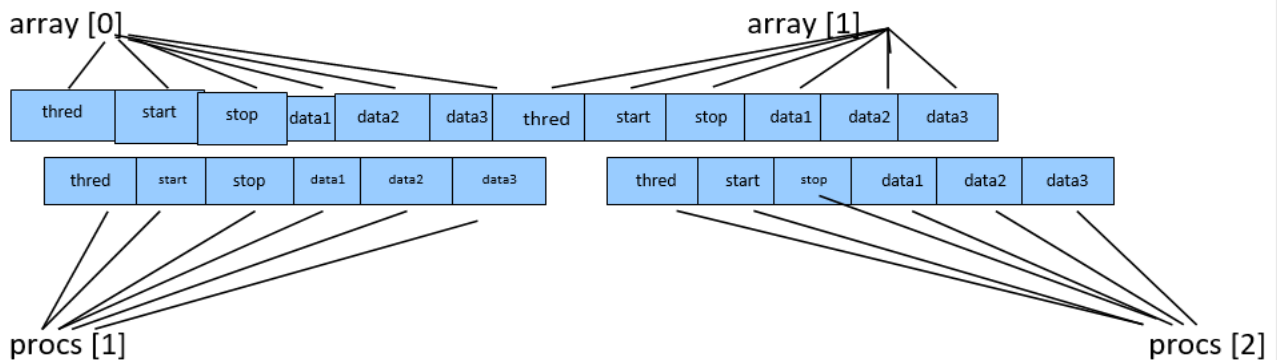


Fig. 1. illustrates the function of MPI_Scatter which makes the scattering of the received parameters. Source: Authors' diagram

3. Results

It is important to identify that the measures are made in three different ways in the performance analysis. The first line in blue refers to the results taken in one node with 8 processors. The second line in red refers to the results taken in two nodes for a total of 16 processors and the last line in green refers to data taken in a total of 144 processors of 18 nodes.¹

It is important to specify that the daemon that initializes the work is MPD which stands for the message-passing daemon. This daemon makes it possible that a copy of the program to be delivered to every node. The way the MPI functions here is in a point-to-point mode where processes communicate with one another. MPI can function in either shared memory or distributed memory. In our case in the media, the shared memory architecture is used.

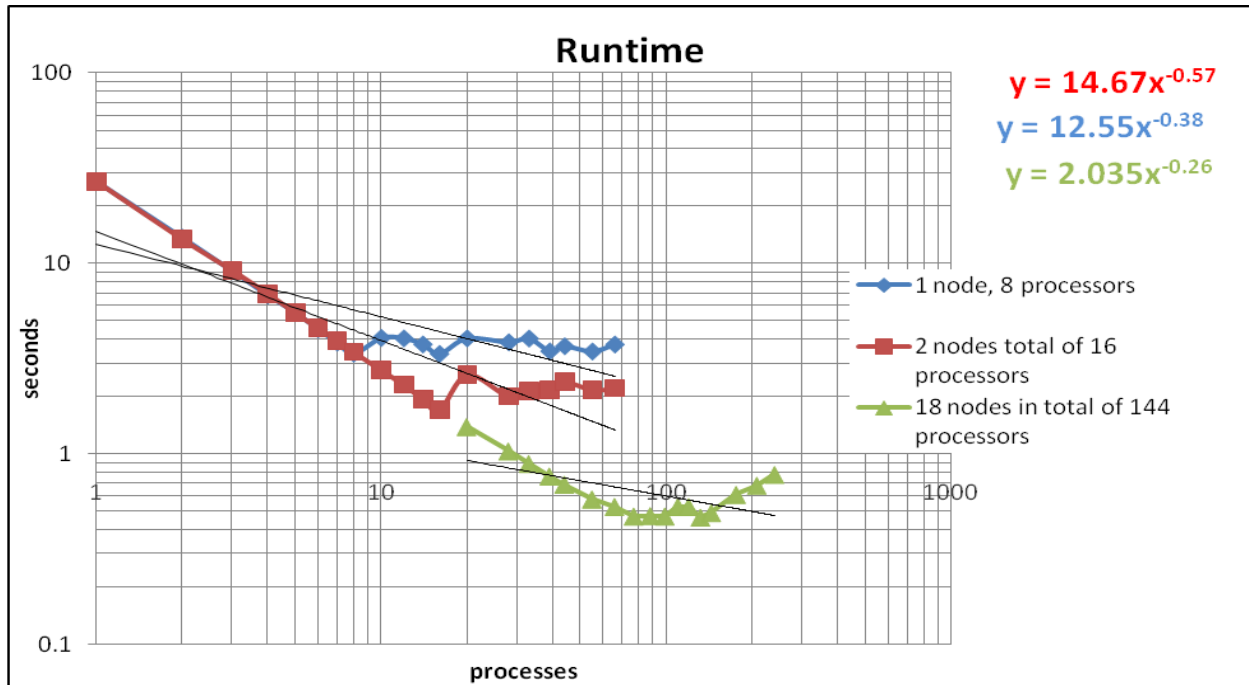


Fig. 2. The relationship between the number of processes and runtime
Source: Authors' calculations

To have a better visual understanding of the problem we have converted the graph axes into log scale, so the descending part is seen clearly. In the three cases, the time decreases until the number of processes reaches the number of processors. This happens because the MPI library and the MPD make it possible that every processor executes one process at a time. The other part of processes that exceeds the number of processors is executed thereafter, resulting in an ascending trend because of swapping overhead. We see that when switching from the line with one process in that of 8, 16 and 144 processes we have rupture points.

The relationship between time and nr of processes as is seen by the trend functions is a power relationship of the class $O(1/x^n)$. In the descending part, we see the fulfillment of theoretical projections that there does exist minimal communication between processes because they are processed in different cores.

4. Discussion

The results of the program execution in the HPC installed in the media remain the same as those we have previously calculated in the academic experimental research of Millo and Frasheri (2013) [5]. Thus, the technological advancements that the final HPC in the media space has, such as better hardware indicators in RAM and Cache memories, did not affect as much as making a considerable difference between the experiments.

The technology and the number of processors and the program executed were the same [9]. We must note that the program taken in our experiment is integrated as a part of an image processing program that relates with the brightness angle of the image translated to led walls, with the following function: as the larger the indicators, the larger the brightness.

5. Conclusions

The analysis showed that the parallel execution time is reduced if it is passed from the execution of one process to more than one process, leading to an increase in performance. We also concluded that the shortage in time is by the number of processors and not that of the processes, however, the average time is reduced for every process, and it does not have an overall result when the number of processes is greater than the number of processors.

In this way, the media can make improvements in the technology by adding up processors, but it must further improve even the process execution so to improve the image quality. If the number of processes exceeds the number of processors, the new costs will be noticeable, and the media shall have a noneffective technology investment.

REFERENCES

- [1] Oxford Dictionary. Access date 12 April 2022
<https://www.oxfordlearnersdictionaries.com/definition/english/parallel-processing>
- [2] Adam Smith (2005) "*Wealth of Nations*" pp. 7-8.
- [3] Parhami. B (2002) "*Introduction to Parallel Processing. Algorithms and Architectures*"
https://doc.lagout.org/science/0_Computer%20Science/2_Algorithms/Introduction%20to%20Parallel%20Processing%20-%20Algorithms%20and%20Architectures.pdf Access date 1 April 2022
- [4] Barnie. B (2012) "*Introduction to Parallel Computing*"
[https://books.google.al/books?id=pmSiDAAAQBAJ&pg=PA466&lpg=PA466&dq=Barnie.+B+\(2012\)+%22Introduction+to+Parallel+Computing%22&source=bl&ots=ytfloMHu7b&sig=ACfU3U1_Q7ixe5dc-pxv77M9gLr8xrzThQ&hl=sq&sa=X&ved=2ahUKewinwOailr3AhU1SvEDHQyfCa8Q6AF6BAGREAM#v=onepage&q=Barnie.%20B%20\(2012\)%20%22Introduction%20to%20Parallel%20Computing%22&f=false](https://books.google.al/books?id=pmSiDAAAQBAJ&pg=PA466&lpg=PA466&dq=Barnie.+B+(2012)+%22Introduction+to+Parallel+Computing%22&source=bl&ots=ytfloMHu7b&sig=ACfU3U1_Q7ixe5dc-pxv77M9gLr8xrzThQ&hl=sq&sa=X&ved=2ahUKewinwOailr3AhU1SvEDHQyfCa8Q6AF6BAGREAM#v=onepage&q=Barnie.%20B%20(2012)%20%22Introduction%20to%20Parallel%20Computing%22&f=false) Access date 01-10 April 2022
- [5] Millo.D and Frasherri. F (2013). "*The economic advantages of parallel computing. Case of a program written in parallel with MPI*" proceedings ISTI 2013 ISBN 978-9928 161-38-3
- [6] Xin Li and Jingjing Guo (2019) "*Research on Ship Data Big Data Parallel Scheduling Algorithm Based on Cloud Computing.*" *Journal of Coastal Research Special Issue No. 94: Selected Topics in Coastal Research: Engineering, Industry, Economy, and Sustainable Development (SUMMER 2019)*, pp. 535-539 (5 pages) Published By: Coastal Education & Research Foundation, Inc.
- [7] Jonathan Zieg and David G. Zawada (2021). "*Improving ESRI ArcGIS Performance of Coastal and Seafloor Analyses with the Python Multiprocessing Module.*" *Journal of Coastal Research Vol. 37, No. 6 (November 2021)*, pp. 1288-1293 (6 pages) Published By: Coastal Education & Research Foundation, Inc.
- [8] Buddy Aswin (2021). "*Fog Computing Resource Management for Video Processing Using Evolutionary Bayesian Optimization and Nondeterministic Deep Reinforcement Learning.*" *Military Operations Research Vol. 26, No. 4 (2021)*, pp. 41-66 (26 pages) Published By: Military Operations Research Society

Baci, N. & Millo, D. (2022). The advantages of parallel processing used to measure the performance of image processing on led walls on a national Albanian media *Global Journal of Computer Sciences: Theory and Research*. 12(1), 55-61. <https://doi.org/10.18844/gjcs.v12i1.7448>

- [9] Xinling Feng, Meng Wan, and Shuang Li (2020). “*The Design of Marine Remote Sensing Image Fusion Model Based on Dynamic Load Balancing Strategy.*” *Journal of Coastal Research SPECIAL ISSUE NO. 103. Global Topics and New Trends in Coastal Research: Port, Coastal and Ocean Engineering (SUMMER 2020)*, pp. 943-946 (4 pages) Published By: Coastal Education & Research Foundation, Inc
-